Test Content

Top of Form

**HTML Form Creation Quiz with Node.js and EJS**

**Objective**

Build a web form using Node.js, Express, and EJS that allows users to submit data. You can choose to handle the form data using either URL Encoded or Multipart form data.

**Instructions:**

**Setup:**

* + Initialize a new Node.js project and install necessary packages (express, body-parser, ejs, multer (if using Multipart)).
  + Setup your server using Express.js and set EJS as your view engine.

**Form Creation:**

* + Create an EJS template for your form.
  + The form should include the following fields:
  + Text input for the user's name.
  + Email input for the user's email address.
  + Password input for the user's chosen password.
  + A dropdown menu for the user to select their favourite colour.
  + A checkbox for the user to agree to the terms and conditions.
  + Add a submit button to the form.

**Server Handling:**

* + Set up a route to display the form when the user navigates to the root URL (GET /).
  + Set up a route to handle the form submission (POST /submit-form).If using URL Encoded, configure your middleware accordingly with body-parser.
  + If using Multipart, configure multer middleware

**Processing Data:**

* + Upon form submission, write server-side code to log all the form data to the console.
  + If a file is uploaded, save it to a directory named uploads.

**Response:**

* + Once the form is processed, redirect the user to a confirmation page that thanks them for their submission.

**Testing:**

* + Test your form by running your Node.js server and navigating to http://localhost:3000.
  + Fill out the form and ensure that the data is logged to the console and the file is saved in the uploads directory if applicable.

**Submission:**

* + Submit your complete project folder compressed in a ZIP file.
  + Include a brief explanation of your form handling choice (URL Encoded vs. Multipart) and why you made this choice.
  + Ensure you have a README file with instructions on how to run your project.

**Grading Criteria:**

* + Functionality of the form (40%)
  + Correct data handling on the server-side (30%)
  + Code organization and readability (20%)
  + Correct and complete answers to the questions (10%)

Bottom of Form